

## **DETAILED ACTION**

### ***Response to Arguments***

1. Applicant's arguments filed 6/30/2010 have been fully considered but they are not persuasive.
  - a. With respect to claim 1, applicant argues that Lindhorst does not disclose identifying a rule for creating an event handler method. The examiner disagrees.
    - i. Applicant argues that markup language is not used to identify a rule. However, Lindhorst discloses using markup language (Column 13, lines 22-64) to identify scriptable tags in HTML. Based on the scriptable tags, event handlers are generated (Column 14, lines 47-55 and column 17, lines 23-32). The rule is taught by Lindhorst determining whether or not an action is a canned action, and reacting accordingly to generate the event handler (Column 17, lines 23-32; column 19, lines 14-29; and column 19, line 61 through column 20, line 10).
    - ii. Applicant argues that the identified rule is not used for creating an event handler method. The examiner disagrees. The event handler method linking an event to an action is explicitly generated based at least on the identified rule (Column 19, lines 27-29 and column 20, lines 6-9).

### ***Claim Objections***

2. Claim 26 is objected to because of the following informalities:

a. Claim 26 recites "a sen/let web application framework object" and is an original claim. Claim 26 originally recited "a servlet web application framework object". For the purposes of examination, claim 26 will be interpreted as originally filed. Appropriate correction is required.

***Claim Rejections - 35 USC § 103***

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

4. **Claims 2, 4-13, 15, 18-20, 22-29, 31-37, 39, 42-50, 52-57, 59, 63-64, 66-70, 72, 75-78, and 162-169** are rejected under 35 U.S.C. 103(a) as being unpatentable over Lindhorst (USPN 6,337,696) in view of Lau (USPN 5,987,247) in further view of Quaelor-Bock (USPN 6,023,271).

5. As per **claim 2**, Lindhorst discloses a computer-implemented method of generating computer code for a web application, comprising:

- a. a computing system receiving an input file, the input file comprising markup language text for a graphical user interface (Column 11, lines 30-40)
- b. the computing system generating an event handler skeleton, wherein generating an event handler skeleton comprises (Column 15, lines 60-64):

iii. the computing system parsing the markup language text (Column 11, lines 41-49)

iv. the computing system identifying an input tag in the markup language text; (Column 13, lines 22-64)

v. the computing system identifying in the markup language text, an attribute within the markup language tag (Column 13, line 22-64)

vi. the computing system identifying in the markup language text a value corresponding to the identified attribute (Column 13, lines 38-56).

The attribute could be INPUT TYPE, for example, and the value could be "text", for example.

vii. the computing system using the identified markup language input tag, the identified attribute, and the identified value to identify a rule for creating an event handler method (Column 19, lines 5-29 and column 19, line 61 through column 20, line 9). The rule is identified as either a canned action, or not a canned action.

viii. the computing system using the identified rule and the identified markup language input tag, the identified attribute, and the identified value to create an event handler method (Column 19, lines 14-29 and column 19, line 61 through column 20, line 9). The event handler method is created from an existing canned action or generated. In either case, the identified tag, attribute, and value are used to identify and create the event handler. Also see paragraph 17, line 23 through column 18, line 26.

c. the computing system receiving event handler methods (Column 19, lines 14-29 or column 12, lines 33-44). The canned action scripts are received, and then any generated scripts are received and rebuilt into an HTML document.

d. Lindhorst fails to disclose the computing system receiving web application business logic objects. However, the examiner maintains that it was well known in the art at the time of the invention to do so, as taught by Lau. Lau discloses importing a business logic design (columns 6, lines 24-29).

It would have been obvious to one of ordinary skill in the art at the time of the invention to import and include business logic in the HTML file before building it. The purpose for doing so would have been to make the web application react according to the overall goals of a corporation.

e. the computing system organizing the application framework code, the web application business logic objects and the event handler methods into application source code (Lindhorst, Column 12, lines 33-44)

f. Lindhorst fails to disclose the computing system binding the web application source code with the input files at runtime. However, the examiner maintains that it was well known in the art at the time of the invention to do so, as taught by Quaelor-Bock. Quaelor-Bock discloses binding GUI components to business objects at runtime (column 3, lines 21-25).

It would have been obvious to one of ordinary skill in the art at the time of the invention to implement the code development system of Lindhorst and Lau with the run time binding of files/inputs as suggested by Quaelor-Bock's teachings, for the purpose of reducing error-prone operations during code development (Quaelor-Bock, column 3, lines 5-10). In addition, runtime binding allows for updating files at any point before execution.

6. With respect to **claim 4**, Lindhorst, Lau, and Quaelor-Bock disclose the method of claim 2. Lindhorst further discloses the web application source code is generated in an object-oriented programming language (Column 3, lines 16-30, JavaScript).
7. With respect to **claims 5-6**, Lindhorst, Lau, and Quaelor-Bock disclose the method of claim 2. Lindhorst fails to disclose wherein the object-oriented programming language is Java or C++. However, Lau discloses Java (column 6, line 34) and C++ (column 3, line 65). It would have been obvious to use Java or C++ to build the web application for the purpose of adding functionality not available to a scripting language.
8. With respect to **claim 7**, Lindhorst, Lau, and Quaelor-Bock disclose the method of claim 2. Lindhorst further discloses determining if the application framework code is available for the web application (Column 19, lines 14-29 and 61-67).

9. With respect to **claim 8**, Lindhorst, Lau, and Quaelor-Bock disclose the method of claim 2. Lau further discloses generating a business logic foundation code Column 6, lines 24-29).

10. With respect to **claims 9-10**, Lindhorst, Lau, and Quaelor-Bock disclose the method of claim 2. Lindhorst further discloses generating a graphical user interface code (Column 12, lines 33-44) and wherein generating a graphical user interface code is based on the input files (Column 11, line 30 through column 12, line 32).

11. With respect to **claims 11**, Lindhorst, Lau, and Quaelor-Bock disclose the method of claim 2. Lindhorst further discloses wherein generating an event handler skeleton is based on the input files (column 11, lines 30-58).

12. With respect to **claims 12-13**, Lindhorst, Lau, and Quaelor-Bock disclose the method of claim 2. Lindhorst fails to disclose compiling the web application source code or interpreting the web application source code. However, Lau discloses using Java (column 6, line 34) or C++ (column 3, line 65) to build a web application. It would have been obvious to use Java or C++ to build the web application for the purpose of adding functionality not available to a scripting language. The examiner takes official notice that C++ is a compiled language and that Java is an interpreted language.

13. With respect to **claim 15**, Lindhorst, Lau, and Quaelor-Bock disclose the method of claim 2. Lindhorst discloses wherein the input files are in HTML format (column 11, lines 37-40).

14. With respect to **claim 18**, Lindhorst, Lau, and Quaelor-Bock disclose the method of claim 2. Lindhorst discloses receiving modified input files (column 12, lines 33-44).

15. With respect to **claim 19**, Lindhorst, Lau, and Quaelor-Bock disclose the method of claim 18. Lau and Lindhorst did not explicitly state the method of claim 18, further comprising compiling the modified input files at runtime. Lau demonstrated that it was known at the time of invention to implement using JAVA (column 6, lines 34). Official Notice is taken that Java technology is known to include a just-in-time compiling system (in other words compiling at run time). It would have been obvious to one of ordinary skill in the art at the time of invention to implement the framework development system of Lau and Lindhorst with run-time compiling as suggested by JAVA's teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to allow for changes/improvements right up until actual use of code.

16. With respect to **claim 20**, Lindhorst, Lau, and Quaelor-Bock disclose the method of claim 19. Quaelor-Bock further discloses binding the web application source code with the compiled modified input files at runtime (Column 3, lines 18-35).

17. With respect to **claim 22**, Lindhorst, Lau, and Quaelor-Bock disclose the method of claim 18. Lau and Lindhorst did not explicitly disclose interpreting the modified input

files at runtime. However, Lau demonstrated that it was known at the time of invention to implement using JAVA (column 6, lines 34). Official Notice is taken that Java technology is known to include a interpretation system. It would have been obvious to one of ordinary skill in the art at the time of invention to implement the development system of Lau and Lindhorst with interpreting code as suggested by JAVA's teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to provide a system of easy programmability (interpretation can be changed quickly on the fly, especially useful in system testing).

18. With respect to **claim 23**, Lindhorst, Lau, and Quaelor-Bock disclose the method of claim 22. Quaelor-Bock further discloses binding the web application source code with the interpreted modified input files at runtime (Column 3, lines 18-36). It would have been obvious to do so for the same reason as for claim 2.

19. With respect to **claim 24**, Lindhorst, Lau, and Quaelor-Bock disclose the method of claim 2. Lau further discloses generating application runtime properties (column 5, lines 39-40).

20. With respect to **claim 25**, Lindhorst, Lau, and Quaelor-Bock disclose the method of claim 2. Lau discloses that it was known at the time of invention to utilize database management systems in business logic (column 8, lines 16-25), but fails to explicitly disclose generating application SQL statements. Official Notice is taken that SQL was known at the time of invention. It would have been obvious to one of ordinary skill in the art at the time of invention to implement the code framework system of Lau with



generating SQL as well. This implementation would have been obvious because one of ordinary skill in the art would be motivated to provide Lau's system with the ability to communicate with as many differing systems/environments as possible and thus increasing flexibility and usability.

21. With respect to **claim 26**, Lindhorst, Lau, and Quaelor-Bock disclose the method of claim 2. Lau further discloses wherein the application framework code comprises an application object and a servlet web application framework object (column 5, lines 15-19).

22. With respect to **claim 167**, Lindhorst, Lau, and Quaelor-Bock disclose the method of claim 2. Lindhorst further discloses determining if an application framework code is available for the web application (Column 19, lines 14-29 and 61-67); and if the application framework code is not available, then generating the application framework code (Column 19, line 61 through column 20, line 9).

23. **Claims 27, 78, and 163-164, and 166** recite substantially similar limitations to claim 2, and are therefore rejected using the same art and rationale set forth above.

24. **Claims 48, 64, and 165** recite substantially similar limitations to claims 2 and 18, and are therefore rejected using the same art and rationale set forth above.

25. **Claim 162** recites substantially similar limitations to claims 2, 18, and 19, and is therefore rejected using the same art and rationale set forth above.

26. **Claims 31 and 52** recite substantially similar limitations to claim 4, and are therefore rejected using the same art and rationale set forth above.

27. **Claims 32, 53, and 66** recite substantially similar limitations to claim 7, and are therefore rejected using the same art and rationale set forth above.

28. **Claims 28, 29, 54, and 67-68** recite substantially similar limitations to claim 8, and are therefore rejected using the same art and rationale set forth above.

29. **Claims 33 and 69** recite substantially similar limitations to claim 9, and are therefore rejected using the same art and rationale set forth above.

30. **Claims 34 and 55** recite substantially similar limitations to claim 10, and are therefore rejected using the same art and rationale set forth above.

31. **Claims 35 and 70** recite substantially similar limitations to claim 11, and are therefore rejected using the same art and rationale set forth above.

32. **Claims 36 and 56** recite substantially similar limitations to claim 12, and are therefore rejected using the same art and rationale set forth above.

33. **Claims 37 and 57** recite substantially similar limitations to claim 13, and are therefore rejected using the same art and rationale set forth above.

34. **Claim 39, 59, and 72** recite substantially similar limitations to claim 15, and are therefore rejected using the same art and rationale set forth above.

35. **Claim 42** recites substantially similar limitations to claim 18, and is therefore rejected using the same art and rationale set forth above.
36. **Claims 43, 49, and 75** recite substantially similar limitations to claim 19, and are therefore rejected using the same art and rationale set forth above.
37. **Claim 44** recites substantially similar limitations to claim 20, and is therefore rejected using the same art and rationale set forth above.
38. **Claims 45, 50, and 76** recite substantially similar limitations to claim 22, and are therefore rejected using the same art and rationale set forth above.
39. **Claim 46** recites substantially similar limitations to claim 23, and is therefore rejected using the same art and rationale set forth above.
40. **Claims 47, 63, and 77** recite substantially similar limitations to claim 26, and are therefore rejected using the same art and rationale set forth above.
41. **Claims 168-169** recite substantially similar limitations to claim 167, and are therefore rejected using the same art and rationale set forth above.
42. **Claims 14, 16-17, 21, 38, 40-41, 58, 60-62, 71, and 73-74** are rejected under 35 U.S.C. 103(a) as being unpatentable over Lindhorst, Lau, & Quaelor-Bock in view of AAPA (applicant's admitted prior art).
43. With respect to **claims 14, 16-17**, Lindhorst, Lau, & Quaelor-Bock disclose the method of claim 2. Lindhorst fails to explicitly disclose wherein the input files are XML,

cHTML or WML format. AAPA demonstrated that it was known at the time of invention to utilize XML and WML (page 3, lines 13-14). Official Notice is take that cHTML was known at the time of invention. It would have been obvious to one of ordinary skill in the art at the time of invention to implement the input files of Lau and Lindhorst with the above formats as found in AAPA's teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to provide as many formats as possible in order to be of use to the largest community of developers possible and thus increase usefulness of the system.

44. With respect to **claim 21**, Lindhorst, Lau, & Quaelor-Bock disclose the method of claim 2. Lindhorst fails to explicitly disclose wherein the modified input files are compiled into DOM objects at runtime. AAPA demonstrated that it was known at the time of invention to compile mark up language files into DOM (page 3, lines 14-16). It would have been obvious to one of ordinary skill in the art at the time of invention to implement the code development system of Lau and Lindhorst with DOM compilation as found in AAPA's teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to provide a easily handled structure for development (AAPA: page 3, lines 16-22).

45. **Claims 38, 40-41; 58, 60-61; 71, 73-74** recite substantially similar limitations to claims 14, 16-17, and are therefore rejected using the same art and rationale set forth above.

46. **Claim 62** recites substantially similar limitations to claim 21, and is therefore rejected using the same art and rationale set forth above.

### ***Conclusion***

Any inquiry concerning this communication or earlier communications from the examiner should be directed to CHRIS NELSON whose telephone number is (571)270-7256. The examiner can normally be reached on Monday to Thursday, 9AM to 5PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Lewis Bullock can be reached on (571)272-3759. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a

Art Unit: 2193

USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/CHRIS NELSON/  
Examiner, Art Unit 2193

/Lewis A. Bullock, Jr./  
Supervisory Patent Examiner, Art Unit 2193